What is claimed is:

1. A system comprising:

a plurality of components, each of the plurality of components comprising data; and

a component assembly engine, wherein the component assembly engine performs the following steps:

receiving one of the plurality of components, the one of the plurality of components comprising a root component;

determining, as a function of the root component, at least one child component of the root component;

receiving another of the plurality of components, the other of the plurality of components comprising the at least one child component;

rendering content as a function of at least one of said at least one child component and said root component.

2. The system of Claim 1 wherein said content includes a hypertext document.

3. The system of Claim 2 wherein said hypertext document is a webpage.

4. The system of Claim 1 further comprising:

a server;

a network, said component assembly engine being communicatively coupled to the network via the server.

5. The system of Claim 1 wherein the server performs the following steps:

receiving a request for a viewable resource, the receiving being via the network;

identifying said root component as a function of the request for the viewable resource;

communicating said root component having been identified to the component assembly engine.

6. The system of Claim 5 further comprising:

a client, the client sending the request for a viewable resource.

7. The system of claim 6 wherein said client receives said content.

8. A method comprising:

receiving one of a plurality of components, the one of the plurality of components comprising a root component, each of the plurality of components comprising data;

determining, as a function of the root component, at least one child component of the root component;

receiving another of the plurality of components, the other of the plurality of components comprising the at least one child component;

rendering content as a function of at least one of said at least one child component and said root component.

9. The method of Claim 8 further comprising:

receiving a request for a viewable resource;

identifying said root component as a function of the request for the viewable resource.

10. The method of Claim 9 further comprising:

sending said content and said portion of said content.

11. A system for assembling a document capable of being electronically rendered, said document being assembled from a plurality of components, said system comprising:

a clickstream database for storing a user's interaction history;

a metadata database 504 for storing product affinity information;

a component database 508 for persistent storage of components;

a storage unit capable of storing a pointer to a component, a component hierarchy data structure, a processed queue data structure, and a feedback buffer data structure;

a request director 206 capable of determining the nature of an incoming request and directing the request responsive to said determination;

an i/o processor 208 connected to said clickstream database, said metadata database 504, and said request director 206, said i/o processor 208 processing requests to modify a metadata database 504 or to download information from said metadata database 504; and

a component assembly engine connected to said clickstream database, said metadata database 504, said component database 508, said storage unit, and said request director 206, and said i/o processor 208, said component assembly engine 210 receiving a request from said request director 206 and assembling said electronically rendered document responsive to said request director 206.

12. A component assembly engine 210 for assembling information, said component assembly engine 210 comprising:

a memory unit capable of storing a pointer to a component, a component hierarchy data structure, a processed queue data structure, and a feedback buffer data structure;

request receiving means for receiving a request for information, said request containing a reference to a root component;

extracting means for extracting said reference from said request, said extracting means connected to said request receiving means and receiving said request therefrom;

loading means for loading a set of components necessary for building a document, said loading means connected to said extracting means and receiving said reference therefrom, said loading means comprising:

index means for indexing components as they are loaded,

indicating means for indicating which component is active,

child component reference extracting means for extracting a child component reference from said active component,

shadow component processing means for processing a shadow component, and

component loading means for loading a component, said component loading means updating said component hierarchy data structure in response to each component loaded; and

processing means for processing a loaded component hierarchy, said processing means comprising:

process indexing means for indicating which component is active, said active component being next to be processed, said process indicating means indicating components to be processed in revere order from the order of loading, and

component processing means for processing said active component, adding an entry for said processed component in said processed queue, generating properties for said processed component, generating updated variables for said processed component, and generating feedback output for said processed component, said feedback output incorporating any feedback output resultant from the processing of any child components of said active component, said feedback output being placed in a feedback buffer, said feedback buffer, said updated variables, and said generated properties being available during processing of subsequent components; and

delivery means connected to said processing means for delivering the contents of said feedback buffer in fulfillment of said request.

13. A computer program product for causing a computer to customizably build a webpage, said computer program product comprising a computer usable medium having computer readable code thereon, said computer program product comprising:

computer readable code means for receiving a request for a webpage;

computer readable code means for extracting a reference to a root component from said request;

computer readable code means for loading a set of components necessary for building a document, said loading means comprising:

computer readable code means for extracting a reference to a child component from a component,

computer readable code means for processing a shadow component, and

computer readable code means for indexing components as they are loaded,

computer readable code means for indicating which component is active,

computer readable code means for extracting a child component reference from said active component,

computer readable code means for processing a shadow component, and

computer readable code means for loading a component and updating said component hierarchy data structure in response to each component loaded; and

computer readable code means for processing a loaded component hierarchy, said processing means comprising:

computer readable code means for indicating which component is active, said active component being next to be processed, said process indicating means indicating components to be processed in revere order from the order of loading, and component processing means for processing said active component, adding an entry for said processed component in said processed queue, generating properties for said processed component, generating updated variables for said processed component, and generating feedback output for said processed component, said feedback output incorporating any feedback output resultant from the processing of any child components of said active component, said feedback output being placed in a feedback buffer, said feedback buffer, said updated variables, and said generated properties being available during processing of subsequent components; and

computer readable code means for delivering the contents of said feedback buffer in fulfillment of said request.

14. A method for assembling a customized displayable document, said method comprising:

1) receiving a request from an entity for a displayable document to be displayed;

2) determining the identity of the entity who generated the request;

3) deciding whether to proceed in response to said determination;

4) continuing to step (1) if said determination indicates said request is not fillable;

5) determining the nature of said request (information upload, information download, or clickthrough);

6) uploading information and returning to step (1) if said determination indicates the nature of said request is information upload, else downloading information and returning to step (1) if said determination indicates the nature of said request is information download;

7) logging said request in a clickthrough database;

8) setting an Active Component Number i equal to 1;

9) setting a Component Total j equal to 1;

10) setting the jth entry of a Parent Array equal to 0;

11) loading a root component indicated in said request from said entity;

12) associating the component pointer at the jth position in a component pointer array with said root component;

13) determining if said root component is a shadow component;

14) executing at least a portion of at least one of the rules element and the code element of said root component if said step (13) of determining determines said root component to be a shadow component;

15) determining if the struct section of the component indicated by the ith component pointer contains any unloaded entries having a child component reference;

16) if said determination is negative and if said Active Component Number i equals 1, continuing to step (28), otherwise:

reading the ith entry of said Parent Array,

setting said Active Component Number i equal to said read ith entry of said Parent Array, and

continuing to step (15);

17) extracting a first unloaded component identifier from the struct section of the component indicated by the ith component pointer;

18) incrementing said Component Total j;

19) setting jth entry in the Parent Array equal to i;

20) loading said component indicated by said first unfilled component identifier;

21) associating the component pointer at the jth position in said component pointer array with said loaded component;

22) setting said Active Component Number i equal to j;

23) determining if said loaded component is a shadow component;

24) continuing to step (26) if said determination of said step (23) indicates said loaded component is a shadow component;

25) repeating steps (15) to (24) until no further references to secondary components are found;

26) executing at least a portion of at least one of the rules element and the code element of the component indicated by the jth position in said pointer array;

27) continuing to step (15);

28) setting said Active Component Number i equal to said component total j;

29) processing the component indicated by the ith component pointer, said processing

producing at least one of the group comprising: combined feedback, combined properties, and combined variables, said combined feedback including the feedback of descendant components of said component indicated by said ith component pointer, said combined properties including the properties of descendant components of said component pointed to by said ith component pointer, said combined variables including variables named in the variables element of descendant components, said combined feedback, said combined properties, and said combined variables being made available for use in processing all parent components of said component pointed to by said ith component pointer;

30) decrementing said Active Component Number i;

31) if said Active Component Number i equals 0, continuing to step (32), otherwise continuing to step (29);

32) supplying the combined feedback produced by the processing of the component at the 1st component pointer in response to said request.

15. A method for assembling a web page comprising:

a) loading a root component;

b) loading all descendant components of said root component;

c) processing all loaded components in reverse order from the order in which said root component and said descendant components were loaded.

16. A method as in claim 15 wherein at least one descendent component is a shadow component.

17. A method as in claim 15, said step (c) of processing including the step of adding said components to a processed queue, said processed queue providing an index of processed components for use during subsequent processing.

18. A data structure for use in electronically rendering a composite document comprising two or more component structures, each said component structure comprising:

a name element,

a data element,

an interface element,

a family element,

a structure element,

a properties element,

a code element,

an activation code element,

a rules element, and

a variables element.

19. A data structure as in claim 18 wherein at least one element of at least one component structure contains a pointer to a corresponding element of another component structure, whereby when said at least one element is processed, the information contained in said corresponding element is used in processing said at least one element.

20. A data structure as in claim 18 wherein at least one element of at least one component

structure contains a pointer to the code element of said at least one component structure, whereby when said at least one component structure is processed, the contents of said at least one element is obtained by execution of code pointed to in said code element of said at least one component structure.

21. A data structure as in claim 18, wherein said components may reside on different internet accessible computer systems.

22. A data structure as in claim 18, wherein at least one of said two or more component structures comprises a shadow component structure, said shadow component structure containing code or rules respectively contained in the code element of said shadow component structure or the rules element of said shadow component structure, at least one of said shadow component code and said shadow component rules when executed causing said shadow component to be resolved into at least one component.